CNN Digit Recognition

```
import tensorflow as tf
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
input_shape = (28,28,1) x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, MaxPooling2D model = Sequential()
model.add(Conv2D(28, kernel_size = (3,3), input_shape = input_shape))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Flatten())
model.add(Dense(128, activation = tf.nn.relu))
from tensorflow.keras.layers import Dropout
model.add(Dropout(0.2)) model.add(Dense(10, activation = tf.nn.softmax))
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
model.fit(x = x_train, y = y_train, epochs = 10)
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn-v0_8')
plt.figure(figsize = (10, 10))
plt.subplot(4,4,1)
image_index = 2853
predict = x_test[image_index].reshape(28, 28)
pred = model.predict(x_test[image_index].reshape(1, 28, 28, 1))
plt.imshow(x_test[image_index].reshape(28, 28), cmap = 'Greys')
plt.title("Predicted Label: " +str(pred.argmax()))
plt.figure(figsize = (10, 10))
plt.subplot(4,4,2)
image_index = 2000
```

```
predict = x_test[image_index].reshape(28, 28)

pred = model.predict(x_test[image_index].reshape(1, 28, 28, 1))
plt.imshow(x_test[image_index].reshape(28, 28), cmap = 'Greys')

plt.title("Predicted Label: " +str(pred.argmax()))

plt.figure(figsize = (10, 10)) plt.subplot(4,4,3) image_index = 1500 predict =
x_test[image_index].reshape(28, 28)

pred = model.predict(x_test[image_index].reshape(1, 28, 28, 1))
plt.imshow(x_test[image_index].reshape(28, 28), cmap = 'Greys')

plt.title("Predicted Label: " +str(pred.argmax()))

plt.figure(figsize = (10, 10)) plt.subplot(4,4,4)

image_index = 1200 predict = x_test[image_index].reshape(28, 28) pred =
model.predict(x_test[image_index].reshape(1, 28, 28, 1))
plt.imshow(x_test[image_index].reshape(28, 28), cmap = 'Greys')

plt.title("Predicted Label: " +str(pred.argmax()))
```